

VŠB – Technická univerzita Ostrava
Fakulta strojní
Katedra robotiky

Aplikace robotického ramene 1500 Robai
Application of the Robai 1500 Robotics Arm

Student:

Pavel Vavřík

Vedoucí diplomové práce:

prof. Dr. Ing. Petr Novák

Ostrava 2016

Zadání diplomové práce

Student: **Bc. Pavel Vavřík**
Studijní program: N2301 Strojní inženýrství
Studijní obor: 2301T013 Robotika
Téma: **Aplikace robotického ramene 1500 Robai**
Application of the Robai 1500 Robotics Arm
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Popište robotické rameno, odvoďte kinematické schéma včetně pohonů.
2. Popište způsob programování/ovládání jednotlivých pohonů.
3. Navrhněte vlastní ovládací software pro rameno, demonstруйте jej.
4. Vyhotovte kompletní dokumentaci.

Seznam doporučené odborné literatury:

- [1] stránky výrobce ramene Cyton Gamma 1500 <<http://www.robai.com/>>.
- [2] Microsoft Visual Studio <<https://www.visualstudio.com/cs-cz/visual-studio-homepage-vs.aspx>>.
- [3] serva Dynamixel <<http://www.trossenrobotics.com/c/robotis-dynamixel-robot-servos.aspx>>.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Dr. Ing. Petr Novák**

Datum zadání: 11.12.2015

Datum odevzdání: 16.05.2016

prof. Dr. Ing. Petr Novák
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne 16. 5. 2016

A handwritten signature in blue ink, consisting of stylized, cursive letters, positioned above a horizontal dotted line.

podpis studenta

Prohlašuji, že

- jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.
- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§35 odst. 3).
- souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové práce, budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě dne 16. 5. 2016



podpis

Jméno a příjmení autora práce:

Pavel Vavřík

Adresa trvalého pobytu autora práce:

Petrov 250, 696 65 Petrov

Poděkování

Chtěl bych poděkovat prof. Dr. Ing. Petru Novákovy a Ing. Zdenkovy Bobovskému Ph.D za poskytnuté konzultace při přípravě mé diplomové práce.

ANOTACE BAKALÁŘSKÉ PRÁCE

VAVŘÍK, P. *Aplikace robotického ramene 1500 Robai : diplomová práce*. Ostrava : VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra robotiky, 2016, 49 s., Vedoucí práce: Novák, P.

Diplomová práce se zabývá tvorbou ovládací aplikace pro robotické rameno Robai Gamma 1500. V úvodu práce je popsáno robotické rameno. V práci je pospaná ovládací aplikace dodávaná výrobcem. V programovacím jazyce C# byly vytvořeny vlastní třídy pro ovládání jednotlivých článků ramene a celého ramene. Pro možnost demonstrace funkčnosti třídy byla vytvořena jednoduchá ovládací aplikace. V práci jsou popsány jednotlivé funkce obou tříd.

ANNOTATION OF BACHELOR THESIS

VAVŘÍK, P. *Application of the Robai 1500 Robotics Arm : Master Thesis*. Ostrava : VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Robotics, 2016, 49 p. Thesis head: Novák, P.

Master thesis is dealing with creating a controlling application of the Robai Gamma 1500 Robotics Arm. The introduction describes the Robotics Arm. In the thesis is described controlling application from the manufacturer. In programming language C# was created classes for controlling each joint and whole Robotics Arm. For Demonstration purpose was created simply an application for controlling a robotic arm. In thesis are described functions in both classes.

Obsah

Seznam použitého značení	8
Úvod	9
1 Popis robotického ramene	10
1.1 Kinematické schéma ramene	11
1.2 Konstrukce ramene Robai Gamma 1500	12
1.2.1 Konstrukce kloubů s převodem	12
1.2.2 Pohon Dynamixel MX-28T	13
1.2.3 Pohon Dynamixel MX-64T	14
1.2.4 Možnosti efektoru	15
1.3 Připojení ramene	15
1.3.1 Připojení ramene k počítači	16
1.3.2 Připojení ramene k netduinu	17
2 Způsob ovládání ramene	18
2.1 Ovládání pomocí aplikace od výrobce	18
2.1.1 Instalace	18
2.1.2 Spuštění aplikace	23
2.1.3 Popis aplikace	24
2.1.4 Nahrávání a projetí požadované trajektorie	26
3 Ovládací software pro rameno	29
3.1 Popis	29
3.2 Vývoj vlastní třídy	30

3.2.1	Popis třídy Robai.cs	31
3.2.2	Popis třídy RobaiClanek.cs	33
3.2.3	Společné ovládací parametry pohonů Dynamixel MX	42
3.2.4	Popis třídy Dynamixel.cs.....	45
3.2.5	Popis knihovny Dynamixel.dll	45
3.2.6	Popis třídy Dynamixel2Netduino	46
4	Závěr	47
5	Zdroje a použitá literatura	48
	Seznam příloh.....	49

Seznam použitého značení

Značka	Veličina/Význam
MR	Mobilní robot
USB	Univerzální sériová sběrnice
UART	Univerzální asynchronní sériové rozhraní
SMPS	Spínaný zdroj
CW	Směr po směru hodinových ručiček
CCW	Směr proti směru hodinových ručiček
C#	Programovací jazyk
PC	Osobní počítač
ID	Identifikační číslo
V	Jednotka napětí
A	Jednotka proudu
g	Jednotka hmotnosti
Kg	Jednotka hmotnosti
mm	Jednotka délky
m/s	Jednotka rychlosti
°C	Jednotka teploty
kΩ	Jednotka odporu
t_{celk} [s]	Celkový čas pohybu
t_z [s]	Čas zrychleného pohybu
t_u [s]	Čas ustáleného pohybu
t_b [s]	Čas brzděného pohybu
v_1	Rychlost pohonu
a	Zrychlené pohonu
s_{celk}	Celková dráha
s_z	Dráha zrychleného pohybu
s_u	Dráha ustáleného pohybu
s_b	Dráha brzděného pohybu

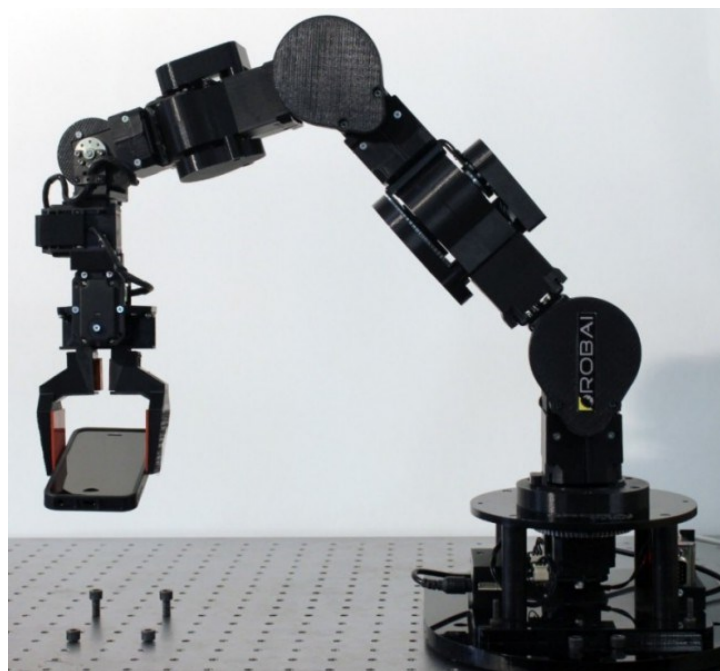
Úvod

Cílem diplomové práce je naučit se pracovat s robotickým ramenem Robai Gamma 1500. Bude popsána mechanické konstrukce ramene.

V práci bude popsána aplikace dodávaná od výrobce. Bude v ní označené ovládání ramene, rozšíření aplikace o rozšiřitelné moduly a ukládání poloh ramene pro možnost dalšího projetí dané trajektorie.

Bude vytvořena a popsána vlastní třída pro ovládání článku a ramene pro Windows i pro Netduino. Pro demonstraci funkčnosti vlastní třídy bude vytvořena a popsána jednoduchá ovládací aplikace.

1 Popis robotického ramene



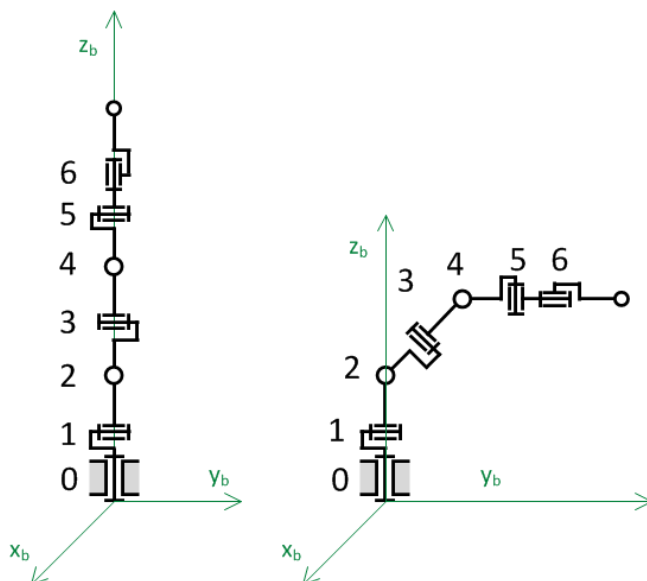
Obr. 1 - Robotické rameno Robai Gamma 1500 [2]

Robai Gamma 1500(nadále rameno) je humanoidní robotické rameno se sedmi stupni volnosti. Rameno je stejně jako lidská ruka kinematicky redundantní, díky tomu dokáže dosáhnout na objekt manipulace nacházejícího se například za překážkou, nebo skrz otvor. Rameno nepotřebuje externí řídicí jednotku, stačí počítač s USB a 12V napájecí zdroj.

Tab. 1 - Parametry robotického ramene Robai Gamma 1500 [1]

Počet stupňů volnosti	7
Nosnost (plné vyložení)	1.5kg
Nosnost (poloviční vyložení)	2kg
Dosah	680 mm
Délka ramene	760 mm
Základna	150×177mm
Přesnost polohování	±0.5mm
Napájení	12V
Maximální lineární rychlost	0.45m/s
Maximální rychlost	0.7m/s
Pracovní prostředí	10°C ÷ 35°C

1.1 Kinematické schéma ramene



Obr. 2 – Kinematické schéma ramene Robai

Rameno obsahuje sedm rotačních kloubů navzájem kolmých. Rozsah jednotlivých os je uveden v Tab. 2.

Tab. 2 - Rozsah pohybu jednotlivých kloubů [1]

ID	Název	Rotace kolem osy	Rozsah	
			Od	Do
0	Základní rotace	Z	-150°	+150°
1	Naklápění ramene	Y	-105°	105°
2	Lámání ramene	X	-105°	105°
3	Naklápění loktu	Y	-105°	105°
4	Lámání loktu	X	-105°	105°
5	Naklápění zápěstí	Y	-105°	105°
6	Rotace zápěstí	Z	-150°	150°
7	Efektor	-	0	35mm

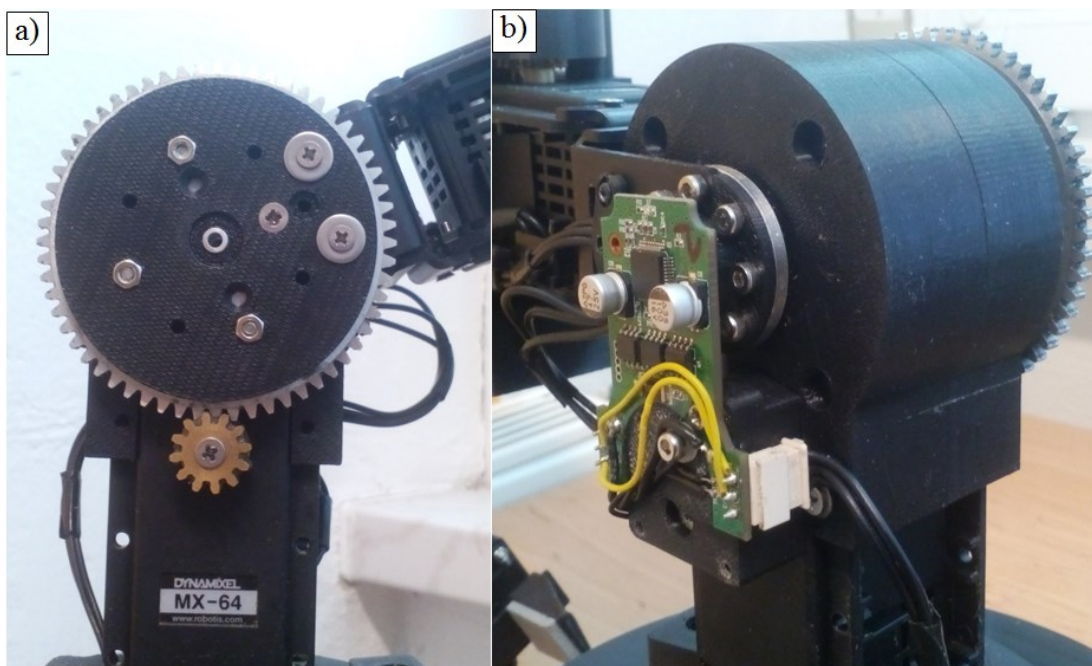
1.2 Konstrukce ramene Robai Gamma 1500

Rameno se skládá z osmy upravených digitálních pohonů Dynamixel od jihokorejské firmy Robotis. Jednotlivé pohony jsou mezi sebou propojeny sériovou linkou UART. Komunikaci mezi počítačem a pohony zajišťuje převodník USB2dynamixel. Prvních pět článků je upraveno a rozšířeno o převodovku. Nosné díly jsou kombinací držáků od firmy Dynamixel a dílů vytištěných na 3D tiskárně.

Tab. 3 - Pohony jednotlivých kloubů

Kloub č.	Pohon	Převod
0	MX-64T	5:1
1	MX-64T	5:1
2	MX-64T	5:1
3	MX-28T	5:1
4	MX-28T	3:1
5	MX-28T	-
6	MX-28T	-
7	MX-28T	35mm/90°

1.2.1 Konstrukce kloubů s převodem



Obr. 3 - Konstrukce kloubů s převodem, a) Pohled na čelní ozubení, b) Pohled na upravenou elektroniku pohonu Dynamixel

Kloub se skládá z upraveného pohonu Dynamixel, ozubeného soukolí, plastového domku, upravené elektroniky a kovového držáku. Pohon výrobce rozebral, vyjmul z něho řídicí elektroniku a vyvedl k ní dva vodiče k připojení motoru Maxon a dva vodiče

k termočlátku, který hlídá teplotu motoru. Pohon složen a osazen ozubeným kolem. Na pohon Dynamixel připevnil plastový domek, který obsahuje ložiska a hřídel ke které je připevněný kovový držák dalšího kloubu a ozubené kolo. Upravená elektronika byla připevněna tak aby zabudovaný magnetický enkóder mohl snímat polohu hřídele za převodovkou.

1.2.2 Pohon Dynamixel MX-28T



Obr. 4 - Pohon Dynamixel MX-28T [7]

Chytrý pohon Dynamixel MX-28T má následující parametry:

[7]

– Převodový poměr	193:1 (Kovové ozubení)
– Absolutní bezkontaktní enkóder	360°/4096
– Komunikace	Half duplex TTL
– Operační napětí	10 až 14,8 V
– Maximální statický moment	2,5 Nm při 12 V
– Maximální proud	1,4 A při 12 V
– Rychlost bez zátěže	55 ot/min při 12 V
– Rozměry	35,6 × 50,6 × 35,5 mm
– Hmotnost	77 g

1.2.3 Pohon Dynamixel MX-64T



Obr. 5 - Pohon Dynamixel MX-64T[8]

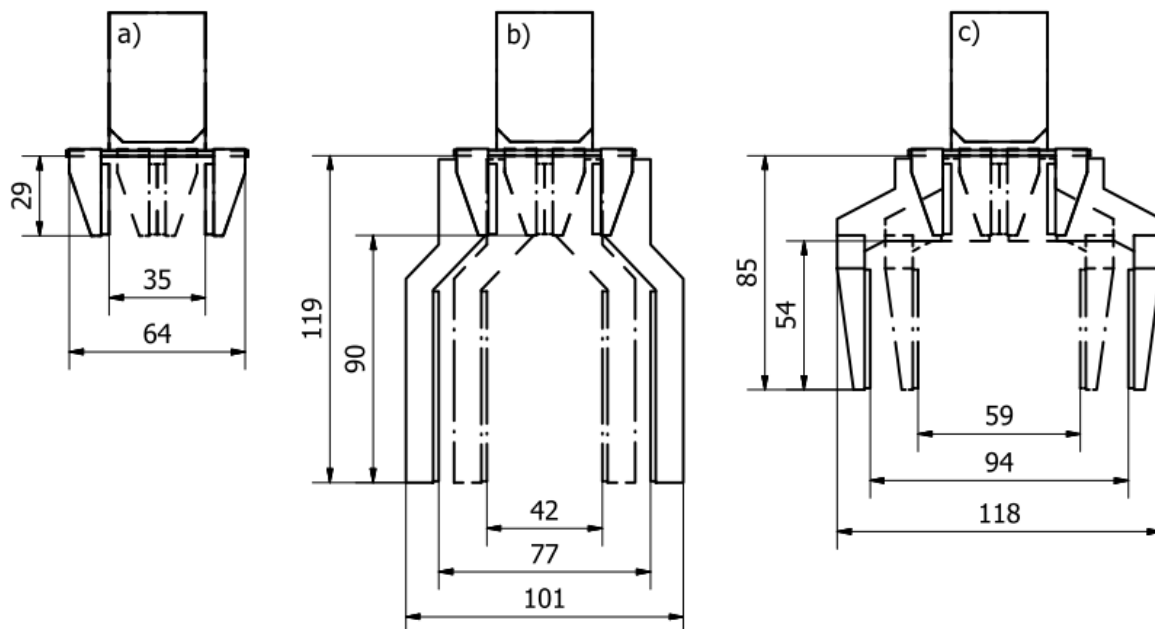
Chytrý pohon Dynamixel MX-64T má následující parametry:

[8]

– Převodový poměr	200:1 (Kovové ozubení)
– Absolutní bezkontaktní	enkóder 360°/4096
– Komunikace	Half duplex TTL
– Operační napětí	10 až 14,8 V
– Maximální statický moment	6 Nm při 12 V
– Maximální proud	4,1 A při 12 V
– Rychlost bez zátěže	63 ot/min při 12 V
– Rozměry	40,2 × 61,1 × 41 mm
– Hmotnost	135 g

1.2.4 Možnosti efektoru

Robotické rameno je vybaveno efektem, který se skládá z pohonu dynamixel MX-28T a pomocí převodu je pohyb převáděn z rotačního pohybu na lineární. Součástí balení jsou dva páry nadstavujících čelistí – dlouhé a široké čelisti.



Obr. 6 - Možnosti přidavných čelistí efektoru: a) základní čelisti - šířka 19mm, b) dlouhé čelisti - šířka 19mm, c) široké čelisti - šířka 40mm

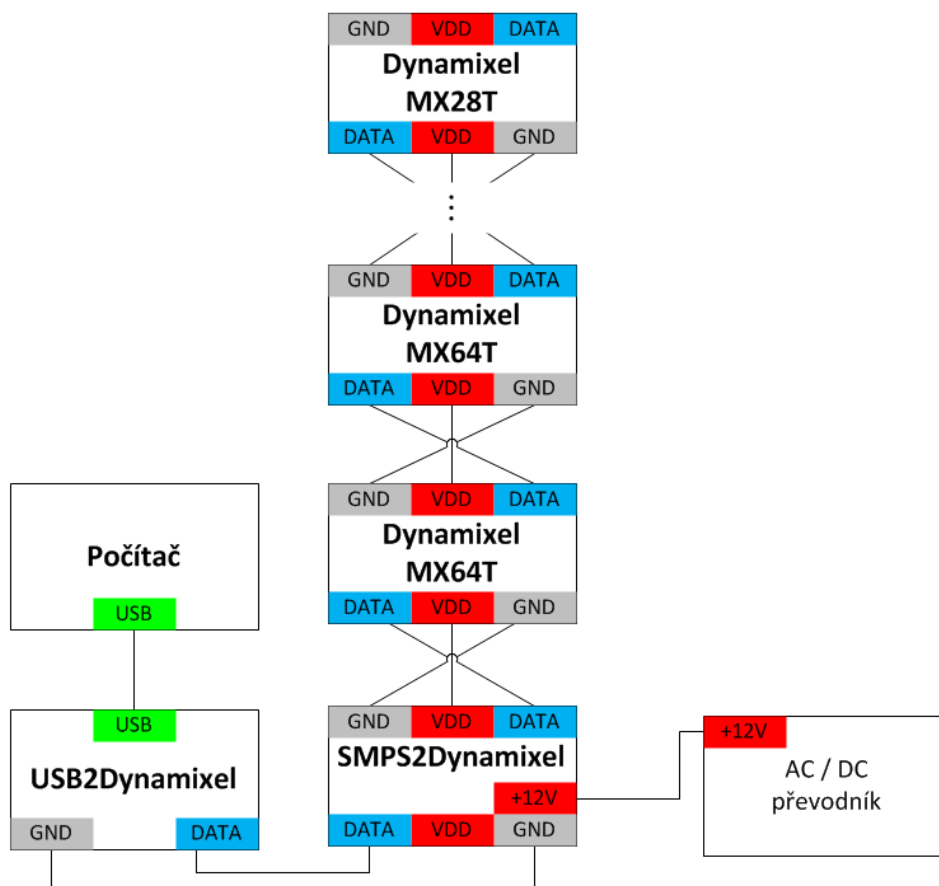
1.3 Připojení ramene

Napájení ramene je zajištěno pomocí adaptéru SMPS2Dynamixel. Adaptér obsahuje dva tři pinové konektory a dva čtyř pinové konektory pro připojení pohonů dynamixel. Jeden 2,5/5,5mm DC jack pro maximální napětí 20V. Pro připojení pohonů Dynamixel MX jsou využity tři pinové konektory.



Obr. 7 - SMPS2Dynamixel [3]

1.3.1 Připojení ramene k počítači

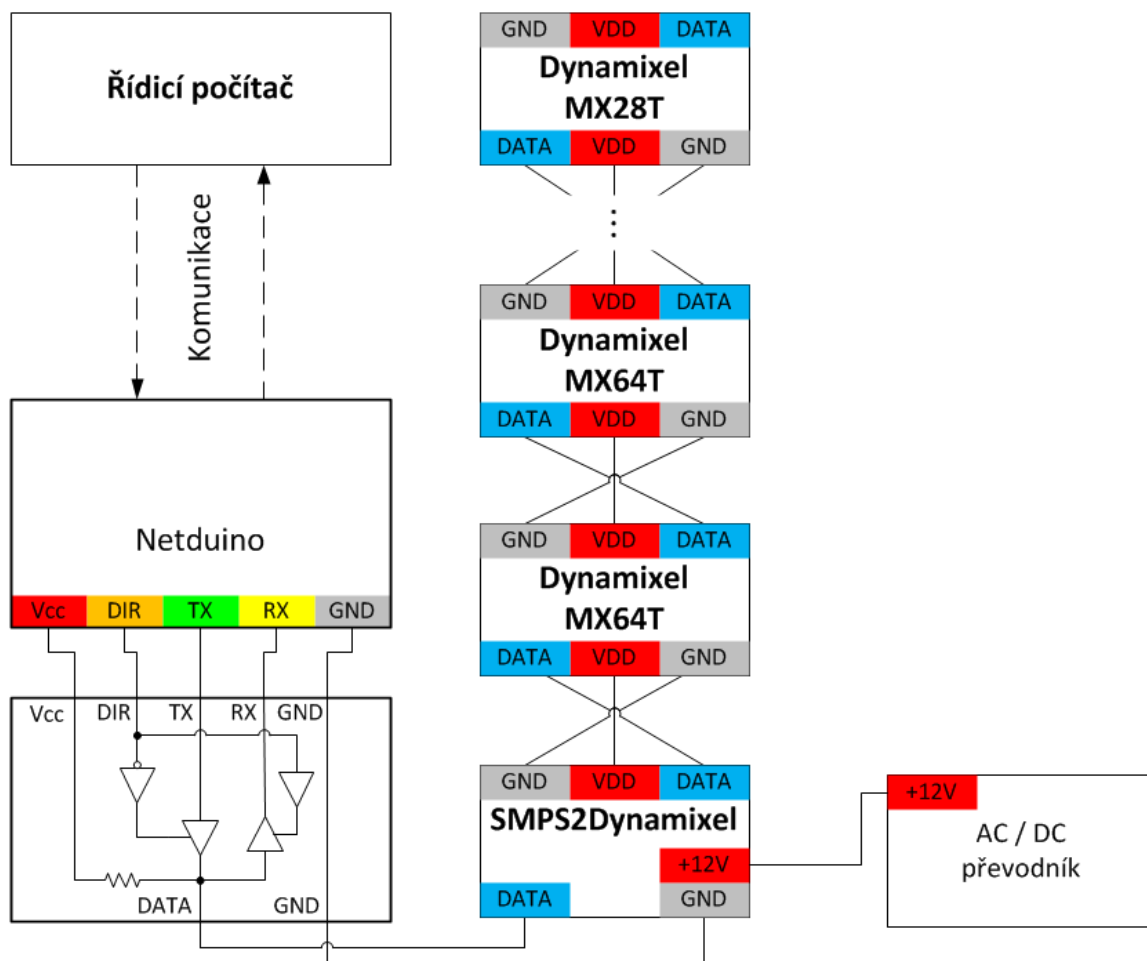


Počítači je připojen k převodníku Dynamixel2USB pomocí USB kabelu. Převodník je napojený na napájecí adaptér pomocí dvou drátů (zem a data). K adaptéru je dále připojené první servo pomocí tří drátů (zem, data, napájení). Stejně jsou propojeny další serva mezi sebou.

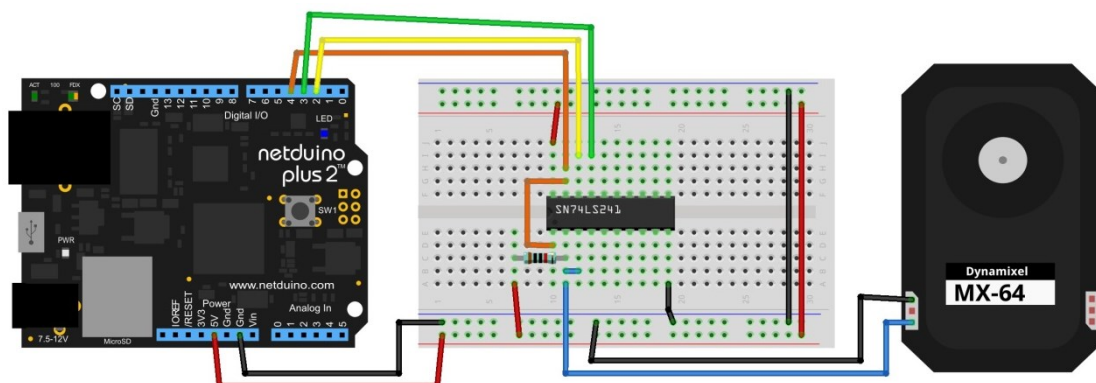


Obr. 8 - USB2Dynamixel převodník [4]

1.3.2 Připojení ramene k netduinu



Netduino je možné použít jako menší řídicí jednotku ramene, díky němuž bude možné použít rameno bez PC, např. jako součást mobilního podvozku, kde Netduino bude komunikovat spolu řídicí jednotkou MR pomocí komunikační sběrnice. Netduino je připojeno přes buffer SN74LS241 s odporem $10\text{k}\Omega$, který se stará o přepínání směru komunikace.



Obr. 9 – Připojení Dynamixelu k Netduino Plus 2

2 Způsob ovládání ramene

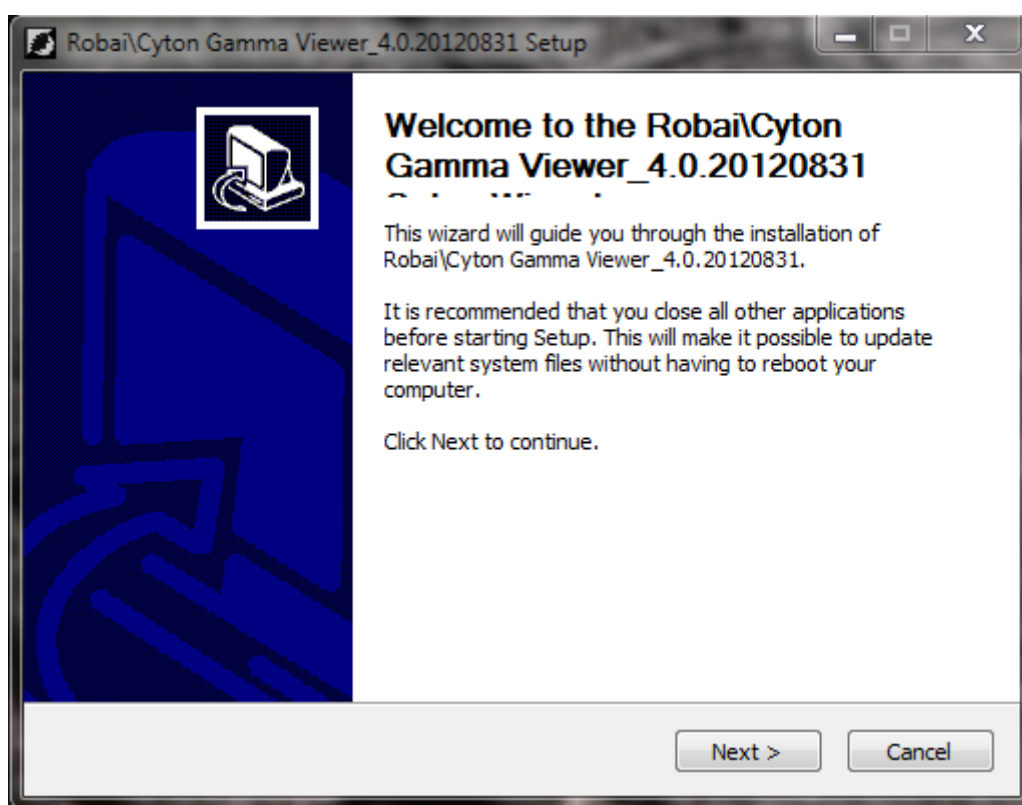
2.1 Ovládání pomocí aplikace od výrobce

Výrobce k rameni dodává ovládací aplikaci Cyton Viewer. Nevýhodou aplikace je ovládání ramene v 3D prostoru pomocí myši ve 2D prostoru. Aplikace obsahuje minimální možnosti, ale nabízí spoustu zásuvných modulů pro rozšíření funkčnosti aplikace.

2.1.1 Instalace

K nainstalování najdeme soubor Cyton Setup. Instalační soubor se nachází na příloženém CD k ramenu Robai a také na CD v příloze diplomové práce.

Po spuštění instalačního souboru se objeví okno instalačního průvodce. Pro pokračování klikneme na Next>.



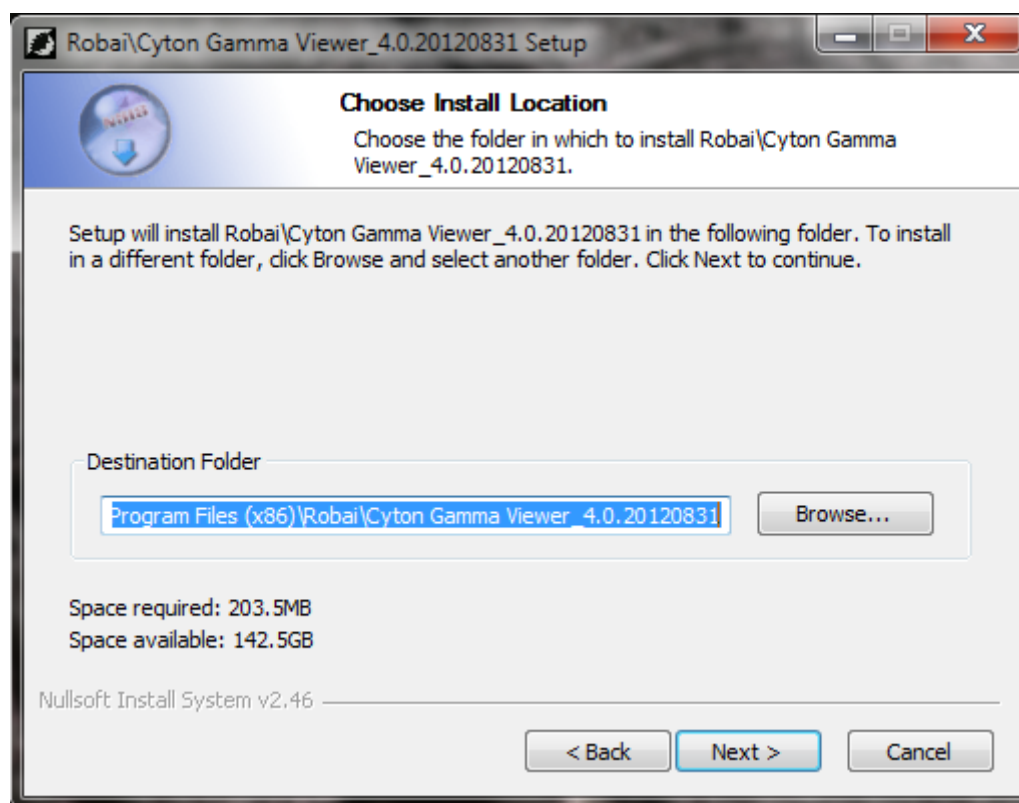
Obr. 10 - Instalační průvodce - úvodní okno

Následně se objeví okno žádající o přečtení a odsouhlasení licenčních podmínek Energid Technologies Corporation pro koncové uživatele. Pro odsouhlasení a pokračování instalace kliknout na I Agree nebo Cancel pro ukončení instalace.



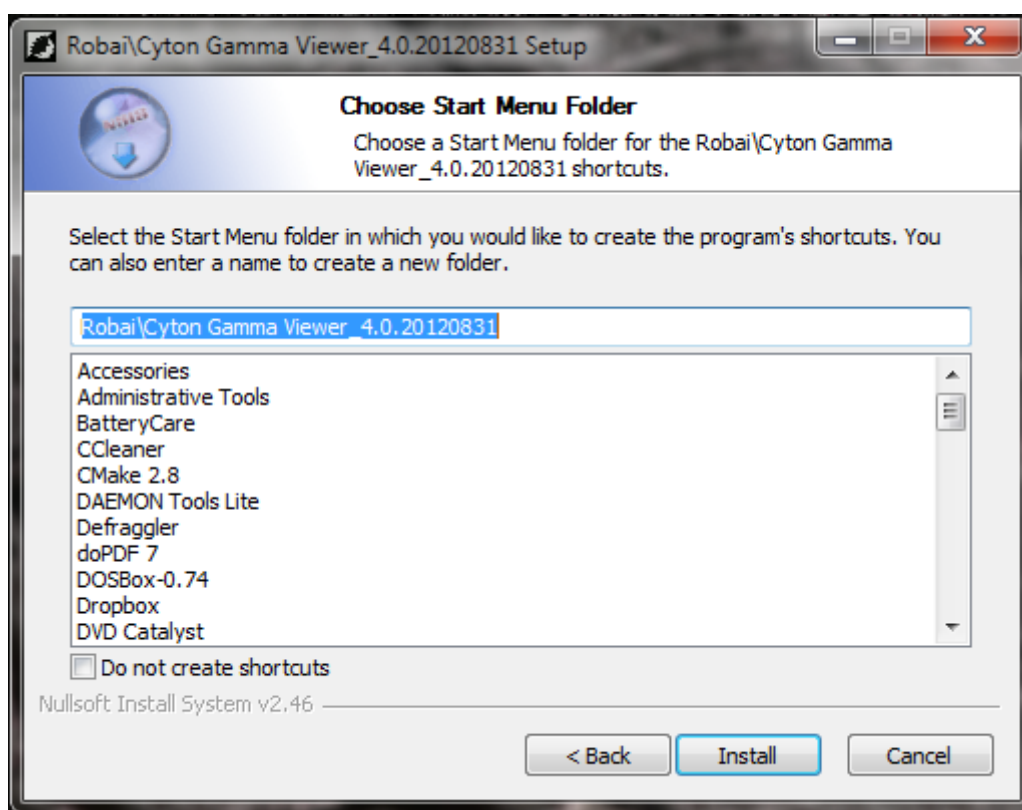
Obr. 11 - Instalační průvodce - licenční podmínky

Následně se objeví okno pro výběr složky, kde bude nainstalován Cyton Viewer. Pro výběr nového místa pro instalaci kliknout na Browse. Pro pokračování instalace kliknout na Next.



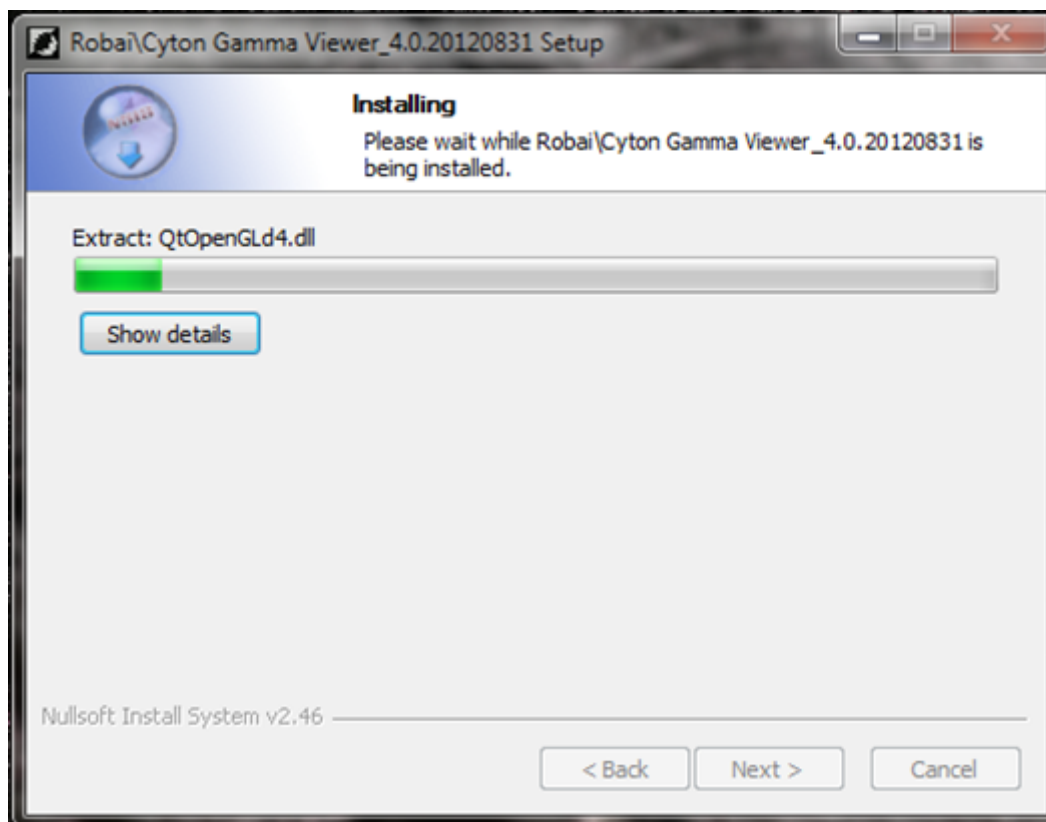
Obr. 12 - Instalační průvodce - výběr umístění

Následně se objeví okno pro výběr umístění zástupce v nabídce Start. Po výběru umístění klikneme na Install pro zahájení instalace softwaru.



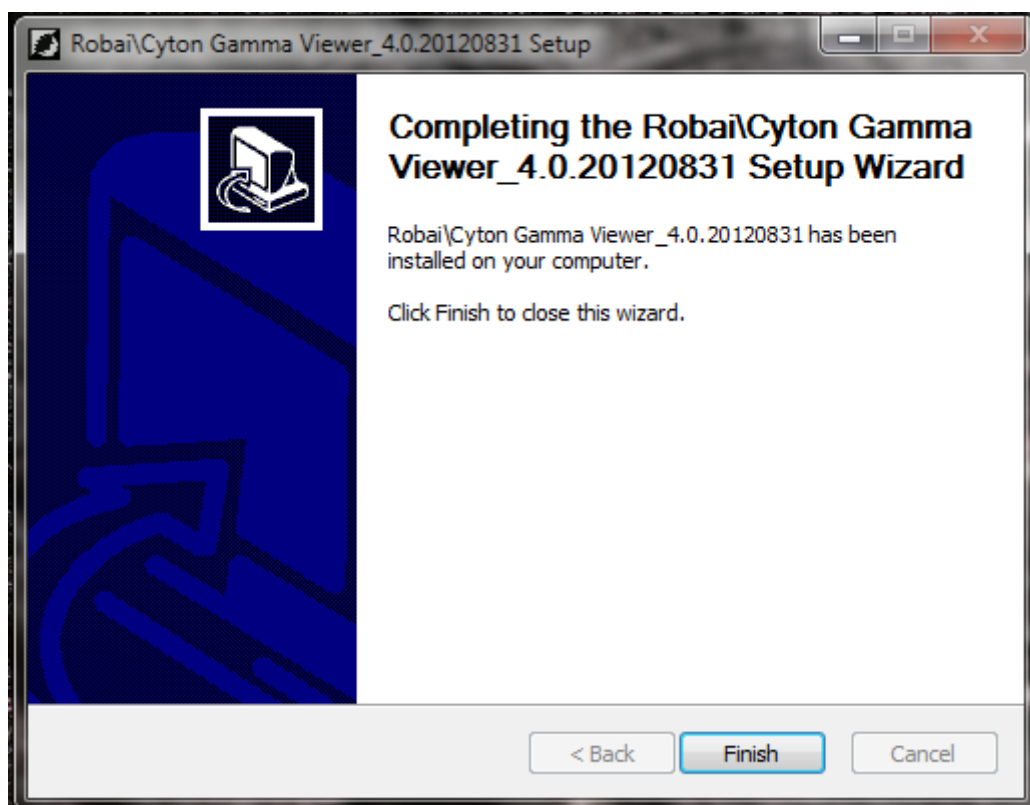
Obr. 13 - Instalační průvodce - výběr umístění v nabídce Start

Zobrazí se okno s průběhem instalace. Pro bližší informace o instalaci klikněte na Show details.



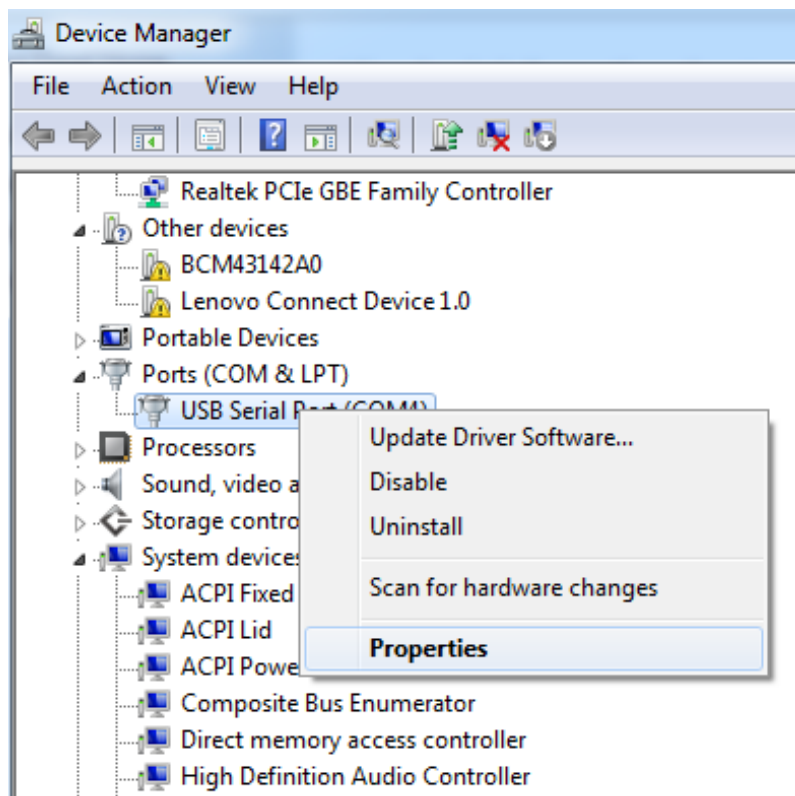
Obr. 14 - Instalační průvodce - průběh instalace

Po dokončení instalace se objeví okno s informací o úspěšném dokončení instalace. Klikneme na Finish pro ukončení průvodce instalace.



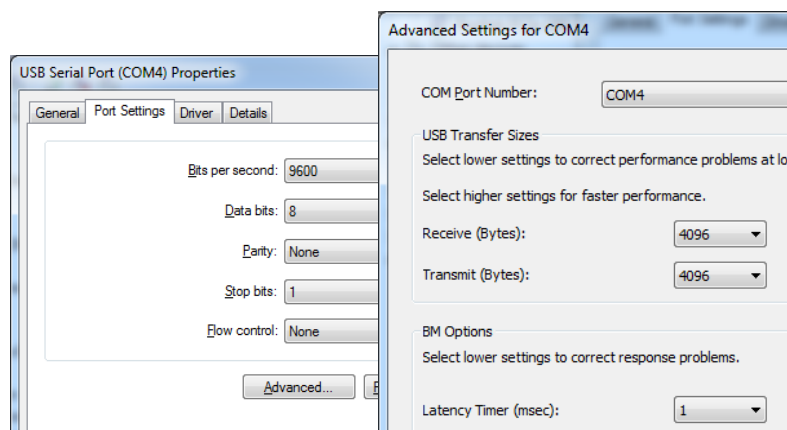
Obr. 15 - Instalační průvodce - dokončená instalace

Po nainstalování je nutné nastavit čas zpoždění. Ve správci zařízení (Tento počítač>Vlastnosti>Správce zařízení) v záložce Porty (COM & LPT) kliknout pravým tlačítkem na „USB Seriál Port“ a zvolit vlastnosti.



Obr. 16 - Vlastnosti portu ve správci zařízení

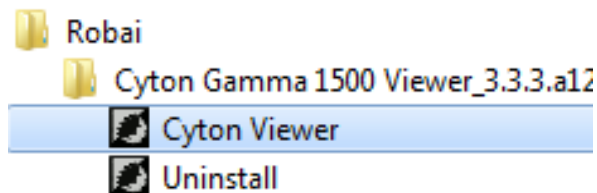
V záložce „Port Settings“ kliknout na „Advance“. V kolonce Latency Timer změnit zpoždění z 16 na 1 mikrosekundu.



Obr. 17 - Změna zpoždění ve vlastnostech portu

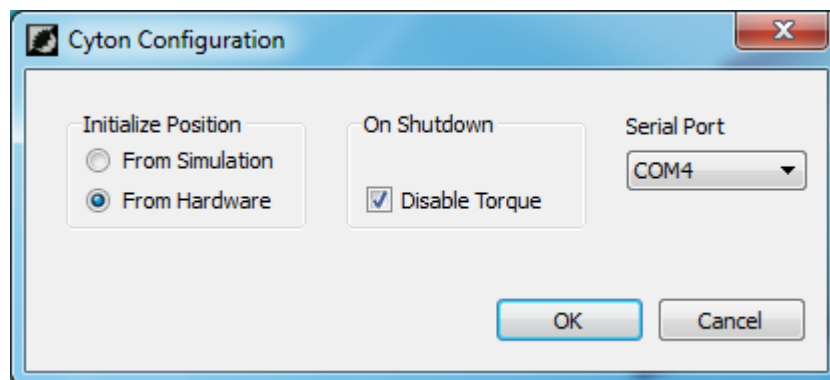
2.1.2 Spuštění aplikace

Před spuštěním aplikace připojíme rameno pomocí USB kabelu k počítači a připojíme napájecí kabel. Pro spuštění aplikace Cyton Viewer najdeme v nabídce Start složku Robai/Cyton Gamma 1500 Viewer/ vybereme a spustíme aplikaci Cyton Viewer.



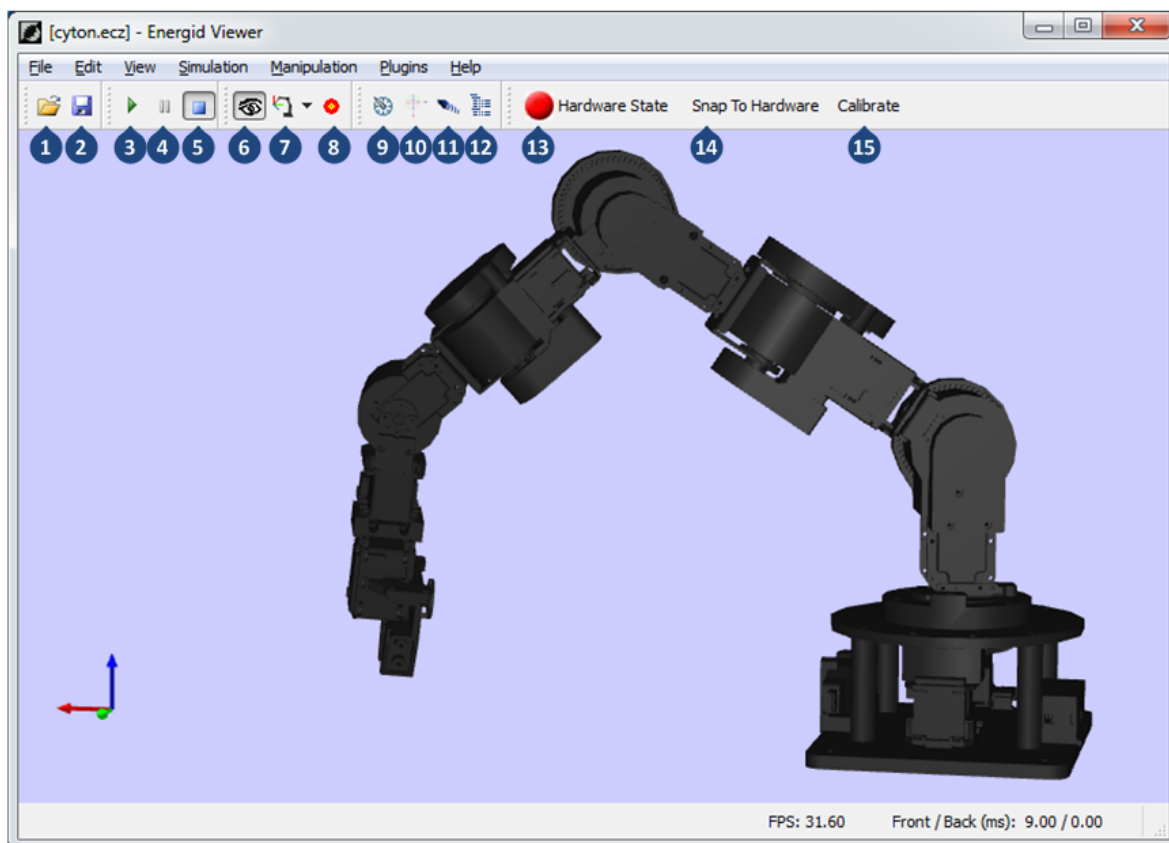
Obr. 18 - Spuštění aplikace Cyton Viewer

Po spuštění aplikace vybereme z nabídky sériový port, na kterém je připojeno rameno Robai. Máme možnost výběru počáteční pozice (Initialize Position). Při počáteční poloze ze simulace (From Simulation) po spuštění se rameno začne pohybovat se všemi klouby do střední polohy. Při počáteční poloze z ramene (From Hardware) se přizpůsobí simulace aktuální poloze ramene a rameno se nepohybuje. Po spuštění jsou sepnuty momenty. Jako poslední možnost nastavení je vypnutí momentu při vypnutí (Disable Torque).



Obr. 19 - Výber sériového portu a počáteční polohy

2.1.3 Popis aplikace



Obr. 20 - Pracovní okno aplikace Cyton Viewer

Popis lišty menu:

- A. **Soubor** (File) – Obsahuje možnosti ukládání a načítání souborů, ukončení aplikace.
- B. **Upravit** (Edit) – Obsahuje možnosti ohledně nastavení aplikace a remena.
- C. **Pohled** (View) – Obsahuje ovládání zobrazení uživatelského okna.
- D. **Simulace** (Simulation) – Obsahuje ovládání simulace – start, stop a pauza.
- E. **Přídavné moduly** (Plugins) – Obsahuje možnost přidání a odebrání přídavných modulů.
- F. **Nápověda** (Help) – Obsahuje informace o aplikaci, odkazy na online nápovědu, ukázkové kódy.

Popis lišty nástrojů:

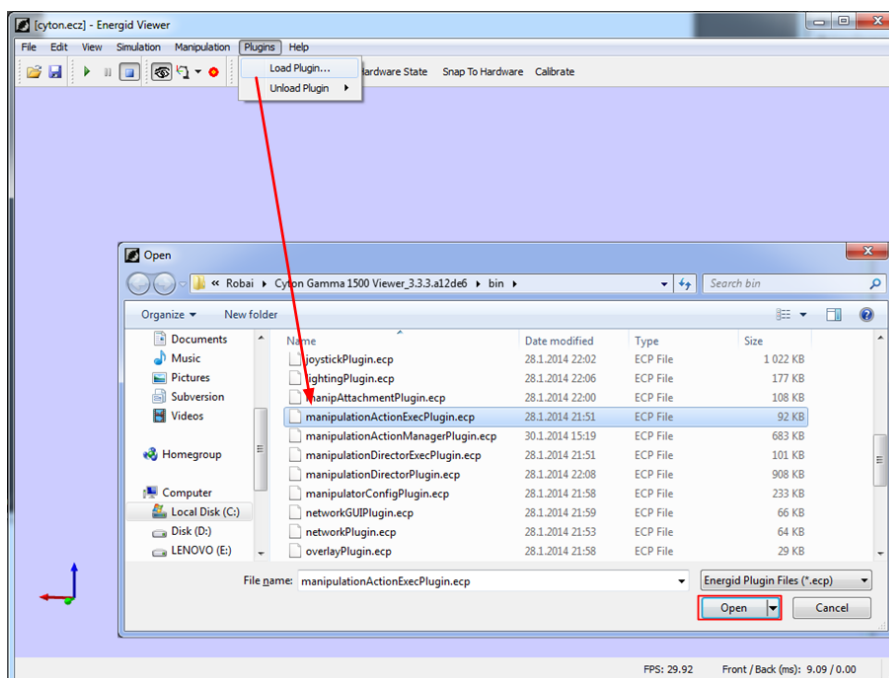
- 1) Otevřít – otevře soubor simulace
- 2) Uložit – uloží soubor simulace
- 3) Spustit simulaci
- 4) Pozastavit simulaci
- 5) Zastavit simulaci

- 6) Režim zobrazení – režim ovládání natáčení, posunování a přiblížení pohledu
- 7) Režim průvodce – režim ovládání koncového efektoru ramene
- 8) Bod zájmu – zobrazí bod ve středu zájmu a umožní možnost ovládání pohledů
- 9) Číselné ovládání ramene – zobrazí okno pro číselné ovládání jednotlivých kloubů
- 10) Posuvník – zobrazení okna pro nastavení posuvníku
- 11) Asistovaný mód – vypnutí pohonů a ovládání ramene ručně
- 12) Kontrola pohonů – zobrazí parametry teploty, zatížení a momentu pro jednotlivé pohony
- 13) Stav připojení ramene – přepne připojení ramene
- 14) Aktualizace modelu simulace – aktualizuje 3D model v simulaci podle polohy ramene
- 15) Kalibrace

Po připojení ramene kdy Stav připojení (13) je zelený v režimu průvodce (7), a je spuštěná simulace (3). Při pohybu myši v pracovním prostoru by mělo rameno následovat model v simulaci. Pokud je simulace zastavená tak při pohybu myši v prostoru se bude pohybovat pouze rameno v simulaci. Po spuštění simulace (3) rameno Robai přejede do polohy ve které je rameno v simulaci.

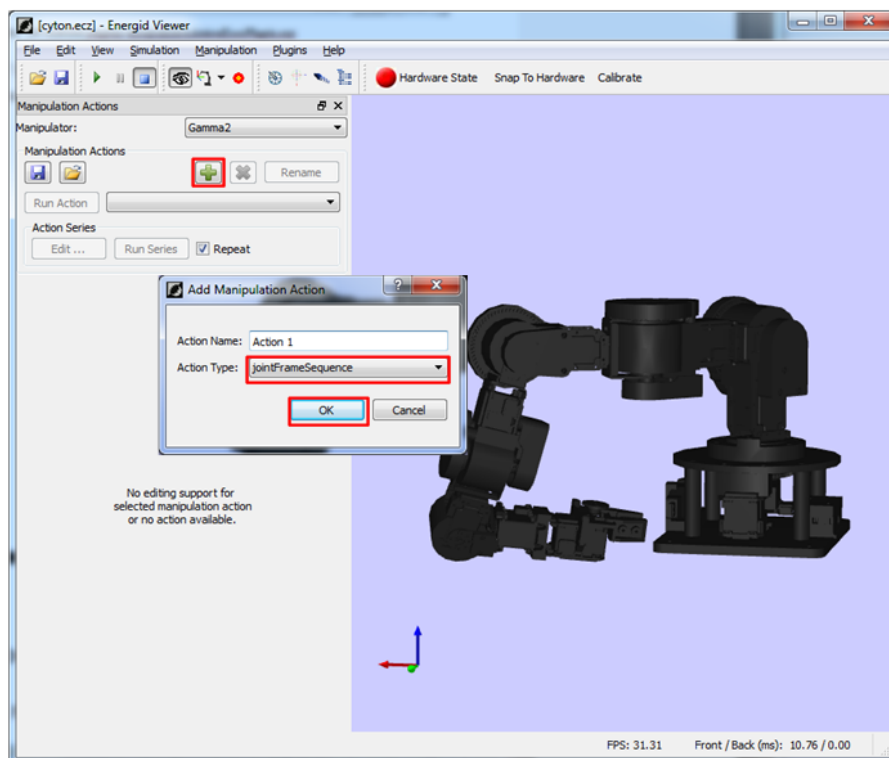
2.1.4 Nahrávání a projektí požadované trajektorie

První je nutné nahrát plugin s názvem manipulationActionExecPlugin.ecp



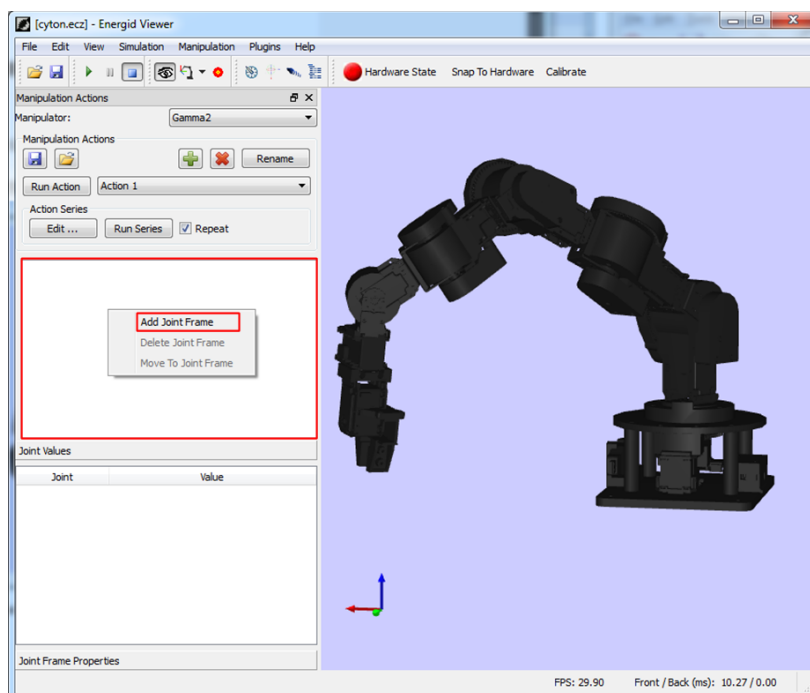
Obr. 21 Nahrání pluginu manipulationActionExecPlugin.ecp

V liště nástrojů se objeví položka Manipulation>Edit Manipulation Actions. Objeví se postranní panel s možností editace nahranných poloh. Přidáme novou akci pomocí kliknutí na zelené plus. Ze seznamu vybereme jointFrameSequence a klikneme na OK.



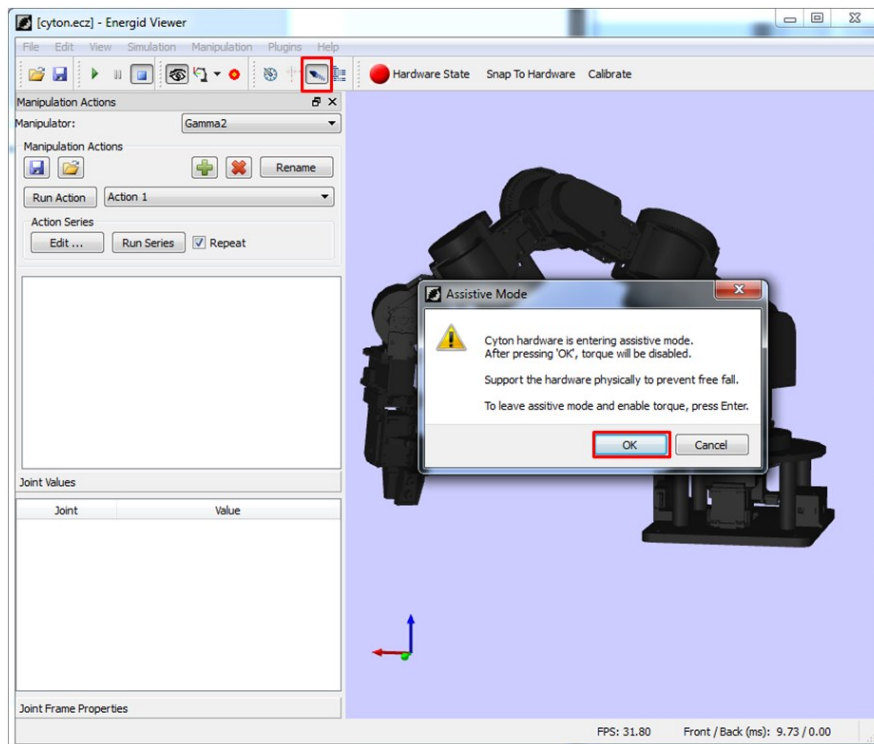
Obr. 22 - Vytvoření nové akce

Ve vyznačené zóně klikneme levým tlačítkem pro uložení nové pozice (Add Joint Frame).



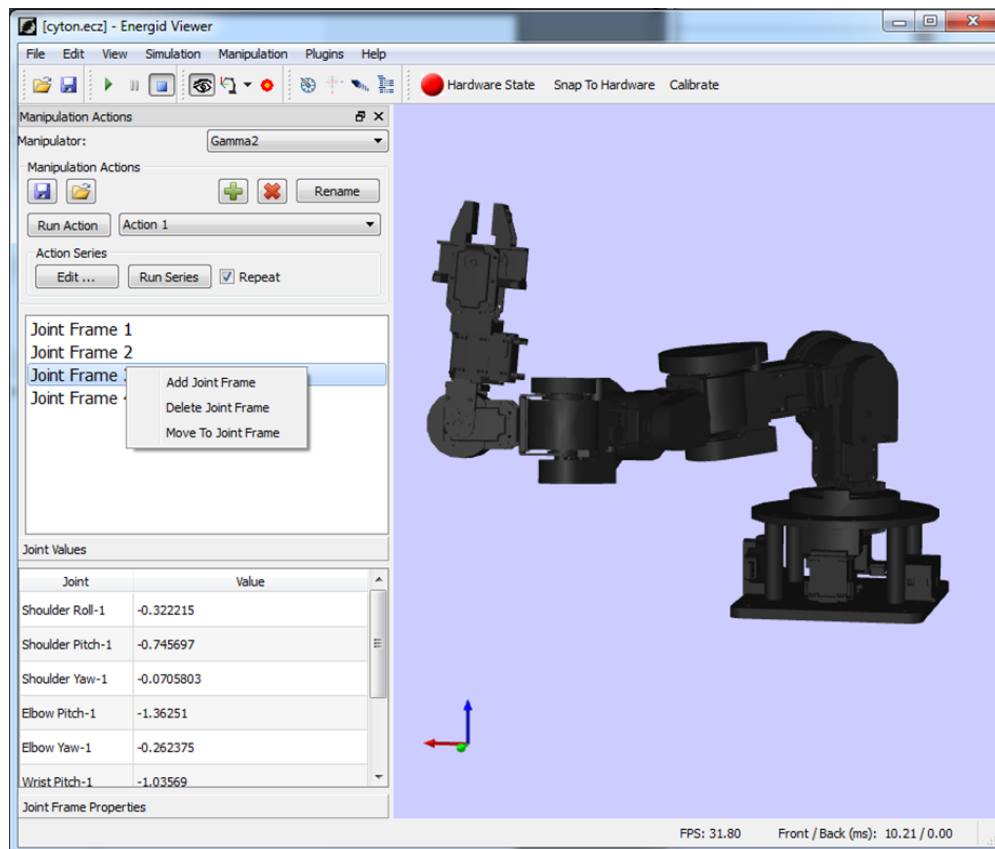
Obr. 23 - Přidání nové polohy kloubů

Pro změnu polohy se přepneme do Asistovaného módu (11). Před potvrzením dialogového okna pomocí OK je nutné držet rameno, protože se vypnou momenty pohonů. Po potvrzení je možné libovolně pohybovat ramenem.



Obr. 24 - Vstup do asistovaného módu ovládání ramene

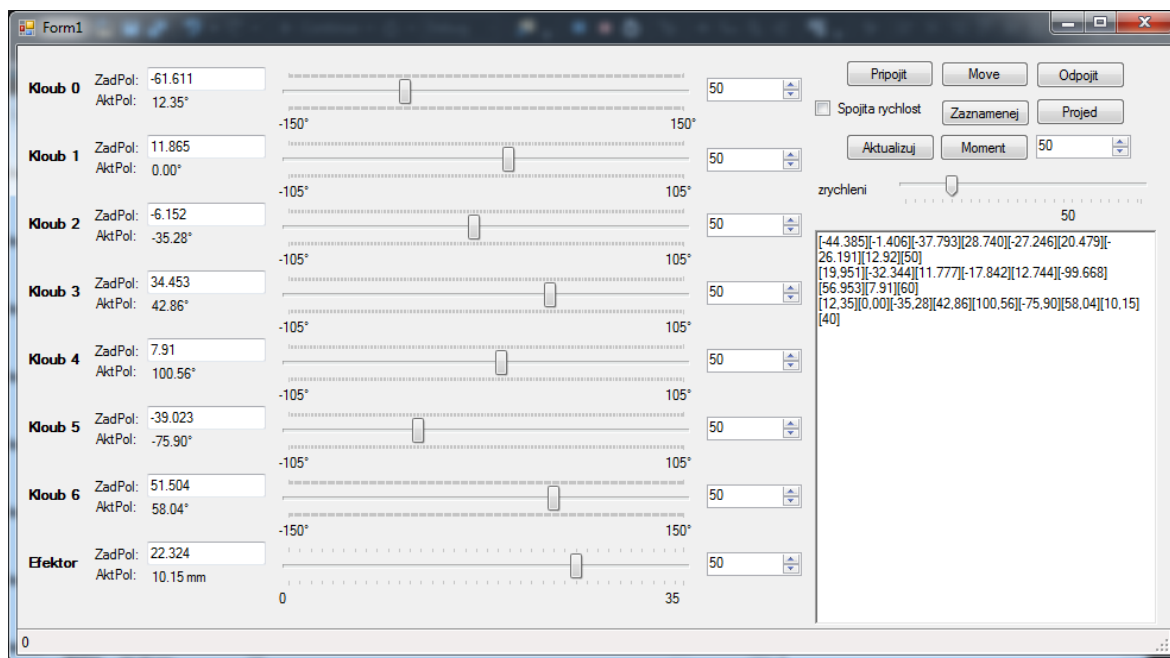
V asistovaném modu se další polohy přidávají pomocí kliknutí pravého tlačítka do bílé oblasti a označení volby Add Joint Frame. Po vytvoření cesty ukončíme asistovaný mód klávesou Enter. Kliknutím pravým tlačítkem myši na Joint Frame, vyvoláme možnost přidat nový bod, odstranit bod a najet ramenem do daného bodu. Bod je možné editovat ve spodní části Joint Values. Uložená trase se projede stisknutím tlačítka Run Action.



Obr. 25 - učení trasy, přidávání nových bodu, mazání špatných a najetí do uloženého bodu.

3 Ovládací software pro rameno

3.1 Popis



Obr. 26 - Vlastní vyvinutá aplikace

Popis ovládacích prvků:

Připojit – připojení ramene robai, vytvoření komunikace, aktualizace parametrů

Odpojit – vypne moment na pohonech, ukončí vzájemnou komunikaci

Move – přesune rameno do poloh zadanych v textovém poli ZadPol

Zaznamenej – uloží polohu ramene

Projed – projede uložené pozice, které jsou zapsané textovém poli

Aktualizuj – aktualizuje posuvníky podle stavu aktuální polohy ramene

Moment – Zapne nebo vypne moment na pohonech

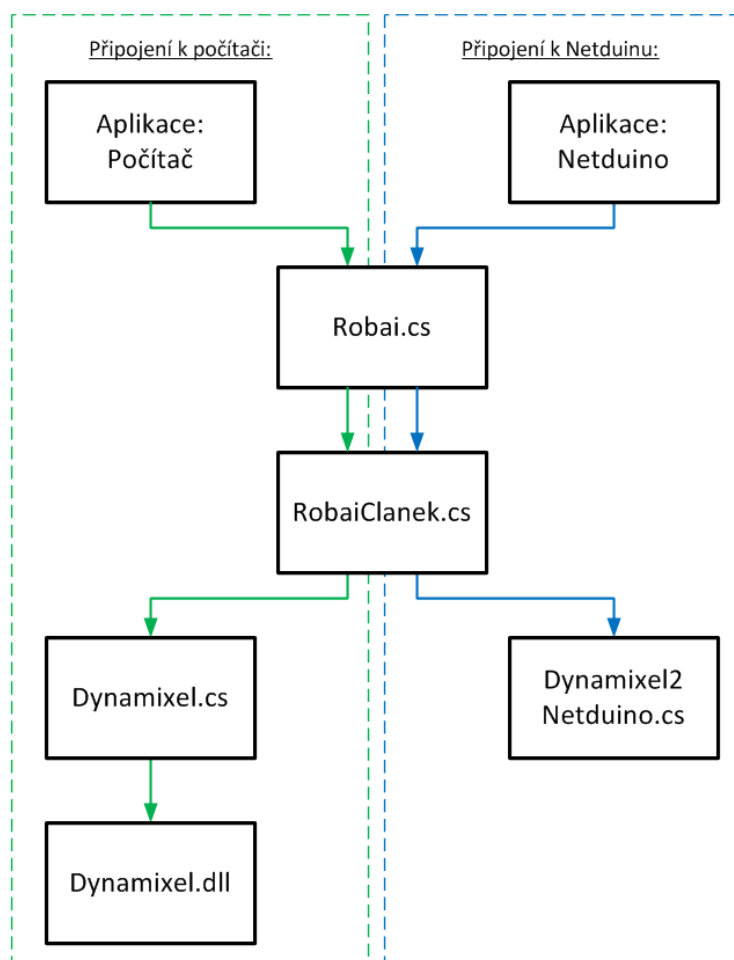
Posuvníky – pomocí posuvníků lze zadat cílovou polohu

Spojité rychlost – Pokud není zaškrtnutá tak se budou pohony pohybovat nastavenou rychlostí, pokud je zaškrtnuta tak se vypočte čas přesunu a podle nejpomalejšího se upraví rychlosti ostatních pohonů

Pomocí posuvníků lze ovládat jednotlivé klouby ramene. Pro učení trasy průjezdu ramene je následující postup:

Připojit rameno k počítači a v aplikaci pomocí tlačítka „Připojit“. Tlačítkem „Moment“ vypnout moment pohonů a pohybovat ramenem do potřebných poloh. V dané poloze pomocí tlačítka „Zaznamenej“ se uloží aktuální poloha ramene spolu s rychlostí, kterou se má do dané polohy dostat. Po uložení všech potřebných poloh průjezdu se tlačítkem „Moment“ opět zapne moment pohonů a rameno bude držet v poslední poloze. Tlačítkem „Projed“ rameno projede uložené polohy.

3.2 Vývoj vlastní třídy



Obr. 27 - Blokové schéma propojení tříd

Struktura vlastní třídy byla volena tak aby základ dal použít i pro programování v aplikaci Windows i v aplikaci pro Netduino. Schéma rozdělení knihovny je na Obr. 27. Soubory Dynamixel.dll a Dynamixel.cs jsou soubory od výrobce a zajišťují komunikaci mezi počítačem a pohony Dynamixel. Soubor Dynamixel.cs byl rozšířený o nové funkce. Soubor Dynamixel2Netduino.cs zajišťuje komunikaci mezi Netduinem a pohony Dynamixel. Soubor RobaiClanek.cs obsahuje strukturu objektu pro jeden článek ramene. Čte a zapisuje jednotlivé parametry pohonu. Soubor Robai.cs obsahuje strukturu objektu

celého ramene Robai Gamma 1500. Vytváří si osm objektů pro každý článek a následně k nim přistupuje podle potřeby.

3.2.1 Popis třídy Robai.cs

Seznam přístupných polí proměnných, které obsahují údaje ze všech pohonů:

- BaudRate[8]** – Informace o kódu přenosové rychlosti
- ReturnDealyTime[8]** – Čas zpoždění před odesláním odpovědi
- CWAngleLimit[8]** – Limit pohonu ve směru hodinových ručiček
- CCWAngleLimit[8]** – Limit pohonu v protisměru hodinových ručiček
- HighestLimitTemperature[8]** – Teplotní limit pohonů
- LowestLimitVoltage[8]** – Minimální přípustné napájení
- HighestLimitVoltage[8]** – Maximální přípustné napájení
- MaxTorque[8]** – Maximální možný moment
- StatusReturnLevel[8]** – Informace o chování návratového statusu
- AlarmLED[8]** – Hodnota chyb při které bude blikat LED dioda
- AlarmShutdown[8]** – Hodnota chyb při kterých se vypne pohon
- TorqueEnable[8]** – Informace o sepnutí momentu
- LED[8]** – Informace o stavu LED diody
- GoalPosition[8]** – Cílová poloha
- MovingSpeed[8]** – Rychlost pohybu do cílové polohy
- TorqueLimit[8]** – Velikost momentu se kterým se pohybuje do cílové polohy
- PresentPosition[8]** – Současná poloha
- PresentSpeed[8]** – Současná rychlost pohybu
- PresentLoad[8]** – Současná velikost proudu odebíraná motorem
- PresentVoltage[8]** – Současná velikost napájení pohonu
- PresentTemperature[8]** – Současná teplota pohonu
- GoalAcceleration[8]** – Cílová velikost zrychlení pohonu
- CasPrejezdu[8]** – Potřebný čas pro přesun článku do cílové polohy

Třída obsahuje stejné funkce jako podtřída RobaiClanek.cs. Funkce pro získání údajů od objektů článků jsou v následujících tvarech:

1) *Public int GetPresentTemperature(int ID)*

Vrátí současnou teplotu jednoho kloubu se zadaným ID, hodnotu také uloží do členu pole PresentTemperature[ID]

2) *Public int[] GetPresentTemperature()*

Vrátí pole osmi hodnot současných teplot všech pohonů. Zároveň hodnoty současných teplot uloží i do pole PresentTemperature[8].

Funkce pro zapsání údajů do objektů článků jsou v následujících tvarech:

1) *Public void SetGoalPosition(int ID, int _hodnota)*

Zapíše _hodnotu do cílové polohy kloubu se zadaným ID.

2) *Public void SetGoalPosition(int[] PoleHodnot)*

Zapíše hodnoty z PoleHodnot do cílové polohy kloubů s odpovídajícím ID.

Robai(int ID)

Konstruktor třídy, vytvoří objekt robai s ID. Volá inicializaci. Vstup: int32 naše ID ramene.

Inicializace()

Vytvoří pole Joint[8] osmi objektů robaiclanek. Při vytvoření nastaví všem objektům následující parametry:

ID – Int32 od 0 po 7. Následně přes tyto ID přistupuje k jednotlivým pohonům

Limity pohybu ramene ve stupních – int32

Limity rozsahu pohonů v krocích – int32 0÷4095

Maximální rychlost pohonů v krocích – int32 0÷1024

Vytvoří devátý objekt Joints s ID 254 pro komunikaci se všemi pohony zároveň.

Connect(int _PortNumb)

Připojení článků a vytvoření komunikace.

Vstup: *int _PortNumb* - číslo komunikačního portu.

Vrací *bool* – TRUE při úspěšném připojení, FALSE pokud se nepodaří připojení

Disconnect()

Odpojení článků.

Aktualizace_Parametru(int ID)

Aktualizuje všechny parametry článku s ID a uloží hodnoty do polí parametrů.

3.2.2 Popis třídy RobaiClanek.cs

Přístupné proměnné objektu ve třídě RobaiClanek.cs:

BaudRate – Informace o kódu přenosové rychlosti

ReturnDealyTime – Čas zpoždění před odesláním odpovědi

CWAngleLimit – Limit pohonu ve směru hodinových ručiček

CCWAngleLimit – Limit pohonu v protisměru hodinových ručiček

HighestLimitTemperature – Teplotní limit pohonů

LowestLimitVoltage – Minimální přípustné napájení

HighestLimitVoltage – Maximální přípustné napájení

MaxTorque – Maximální možný moment

StatusReturnLevel – Informace o chování návratového statusu

AlarmLED – Hodnota chyb při které bude blikat LED dioda

AlarmShutdown – Hodnota chyb při kterých se vypne pohon

TorqueEnable – Informace o sepnutí momentu

LED – Informace o stavu LED diody

GoalPosition – Cílová poloha

MovingSpeed – Rychlost pohybu do cílové polohy

TorqueLimit – Velikost momentu se kterým se pohybuje do cílové polohy

PresentPosition – Současná poloha

PresentSpeed – Současná rychlost pohybu

PresentLoad – Současná velikost proudu odebíraná motorem

PresentVoltage – Současná velikost napájení pohonu

PresentTemperature – Současná teplota pohonu

GoalAcceleration – Cílová velikost zrychlení pohonu

CasPrejezdu – Čas který potřebuje kloub pro přesun do cílové polohy

RobaiClanek()

Konstruktor třídy, vytvoří objekt článek a uloží mu hodnoty parametrů do proměných, vstupy:

- *int _ID* – celé číslo, ID pohonu dynamixel 0÷7.
- *int _CCWPosLimit; int _CWPosLimit* – celé číslo, krajní hodnoty chodu ramene v úhlových stupních
- *int _CCW_Angle_Limit; int _CW_Angle_Limit* – celé číslo, krajní hodnoty pohonu v bitových jednotkách 0÷4095
- *int _MaxSpeed* – celé číslo, maximální rychlost pohonu za převodovkou

Connect(int _PortNumb)

Vytvoření komunikace s dynamixeli.

Vstup: *int _PortNumb* - číslo komunikačního portu.

Vrací *bool* – TRUE při úspěšném připojení, FALSE pokud se nepodaří připojení

Disconnect()

Ukončí komunikaci s pohony dynamixel.

GetBaudRate()

Vrátí číslo přenosové rychlosti komunikace. Přenosová rychlost v bitech za sekundu je v Tab. 4.

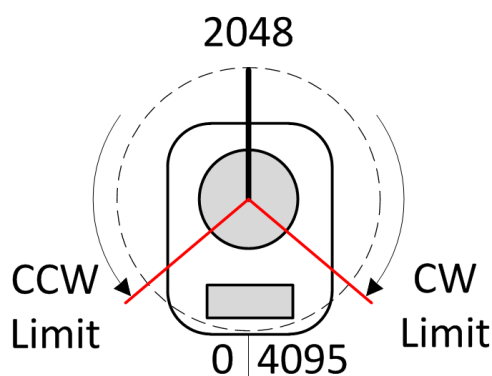
Tab. 4 - Přenosové rychlosti pohonů

Číslo	Dosažená přenosová rychlost	Nastavená přenosová rychlost	Tolerance
1	1 000 000,0	1 000 000	0,000%
3	500 000,0	500 000	0,000%
4	400 000,0	400 000	0,000%
7	250 000,0	250 000	0,000%
9	200 000,0	200 000	0,000%
16	117 647,1	115 200	-2,124%
34	57 142,9	57 600	0,794%
103	19 230,8	19 200	-0,160%
207	9 615,4	9 600	-0,160%
250	2 250 000,0	2 250 000	0,000%
251	2 500 000,0	2 500 000	0,000%
252	3 000 000,0	3 000 000	0,000%

GetReturnDelayTime()

Vrátí čas zpoždění mezi koncem přijetí paketu s instrukcí a odesláním návratového statusu v mikrosekundách. Hodnoty 0÷254 v mikrosekundách, zpoždění je dvakrát větší než nastavená hodnota. Např. při hodnotě 10 je zpoždění 20 mikrosekund.

GetCWAngleLimit(), GetCCWAngleLimit()



Obr. 28 - Okrajové podmínky polohy pohonu

Vrátí nastavenou hodnotu krajních limitů polohy. Int 0÷4095, nejmenší krok odpovídá 0,088°.

GetHighestLimitTemperature()

Vrátí nastavenou hodnotu teplotního limitu ve °C. Po překročení se vyskytne chyba přehřátí.

GetLowestLimitVoltage(), GetHighestLimitVoltage()

Vrátí nastavený operační limit napětí. Rozsah je od 50 do 160 s jednotkou 0,1V. Např. hodnota 120 odpovídá 12V. Pokud vstupní napětí překročí nastavený rozsah, vyskytne se chyba vstupního napětí.

GetMaxTorque()

Vrátí maximální moment pohonu. Hodnoty jsou 0÷1023 s jednotkou 0,1%. Např. hodnota 1023 znamená, že pohon bude využívat 100% dostupného momentu. Při hodnotě 512 bude pohon využívat pouze 50% dostupného momentu.

GetStatusReturnLevel()

Vrátí hodnotu, jak se bude chovat vracet statusový paket.

Tab. 5 - Chování vracení statusového paketu

Hodnota	Návrat statusového paketu
0	Pohon nebude odpovídat na všechny příkazy (kromě příkazu PING)
1	Pohon bude odpovídat pouze na příkaz READ
2	Pohon bude odpovídat na všechny příkazy

GetAlarmLED()

Pohony dynamixel se dokáží chránit sami před poškozením. Funkce vrátí hodnotu, při které bude informační LED dioda blikat při detekci chyby.

Tab. 6 - Popis jednotlivých chyb pohonu dynamixel

Bit	Název	Poznámka
Bit 7	-	-
Bit 6	Chyba instrukce	Nastane, když je přenesena nedefinovaná instrukce, nebo instrukce Akce bez uložené hodnoty v zapisovacím registru
Bit 5	Chyba přetížení	Nastane, když je překročený limit proudu
Bit 4	Chyba kontrolního součtu	Nastane, když nesedí kontrolní součet na konci přijatého paketu
Bit 3	Chyba rozsahu	Nastane, když hodnota přijatého příkazu je mimo daný rozsah použití
Bit 2	Chyba přehřátí	Nastane, když vnitřní teplota překročí nastavený teplotní limit
Bit 1	Chyba úhlového limitu	Nastane, když cílová hodnota je mimo nastavené úhlové limity
Bit 0	Chyba vstupního napájení	Nastane, když je vstupní napájení mimo zadaný rozsah

Například hodnota chyby 0x05 (binárně 00000101) znamená chybu vstupního napájení a chybu přehřátí.

GetAlarmShutdown()

Vrací hodnotu chyby, při které se vypne moment pohonu. Popis chyby v Tab. 6.

GetTorqueEnable()

Vrátí 1, pokud je moment pohonu aktivovaný. Vrátí 0, pokud je moment pohonu deaktivovaný.

GetLED()

Vrátí 1, pokud LED svítí. Vrátí 0, pokud LED nesvítí.

GetGoalPosition()

Vrátí nastavenou cílovou polohu. Hodnota je typu int32 v rozsahu 0÷4095. Jeden dílek odpovídá 0,088°.

GetMovingSpeed()

Vrátí nastavenou rychlost, se kterou se má pohon dostat do cílové polohy. Hodnota je typu int32 v rozsahu 0÷1023.

GetTorqueLimit()

Vrátí hodnotu nastaveného limitu momentu. Hodnota je typu int32 v rozsahu 0÷1023. Jeden dílek je zhruba 0,1%. Např. hodnota 1023 znamená, že pohon využije 100% dostupného momentu.

GetPresentPosition()

Vrátí aktuální polohu pohonu dynamixel. Hodnota je typu int32. Rozsah polohy je 0÷4095. Jeden dílek odpovídá 0,088°.

GetPresentSpeed()

Vrátí aktuální rychlost pohonu dynamixel. Hodnota je typu int32. Rozsah rychlosti je 0÷1023 v proti směru hodinových ručiček a 1024÷2047 ve směru hodinových ručiček.

GetPresentLoad()

Vrátí aktuální velikost proudu působící na motor. Hodnota je typu int32. Rozsah proudu je 0÷1023 v proti směru hodinových ručiček a 1024÷2047 ve směru hodinových ručiček.

GetPresentVoltage()

Vrátí aktuální velikost vstupního napětí. Hodnota je 10 větší než skutečné napětí. Např. hodnota 120 odpovídá 12V.

GetActualTempature()

Vrátí aktuální teplotu ve stupních celsia. Hodnota dat přímo odpovídá aktuální teplotě pohonu.

GetMoving()

Vrátí 1, pokud se pohon pohybuje. Vrátí 0, pokud pohon stojí-

GetGoalAcceleration()

Vrátí cílové zrychlení. Hodnota je typu int32. Rozsah zrychlení je 0÷254.

SetBaudRate(int baudrate)

Nastaví přenosovou rychlost komunikace. Vstup je hodnota typu int32 hodnota je podle Tab. 4.

SetReturnDelayTime(int RDT)

Nastaví čas návratu mezi přijetím paketu s instrukcemi a odesláním statusu paketu. Hodnota je typu int32, jeden díl odpovídá 0,5 mikrosekundám.

SetCWAngleLimit(int CWAngleLimit), SetCCWAngleLimit(int CCWAngleLimit)

Nastaví krajní polohy pohonu. Vstupní hodnota je typu int32, rozsah je 0÷4095. Jeden dílek odpovídá 0,088°. Viz Obr. 28.

SetLowestLimitVoltage(int lowVoltage), SetHighestLimitVoltage(int highVoltage)

Nastaví operační limit napětí. Vstupní hodnota je typu int32. Rozsah je od 50 do 160. Jeden dílek odpovídá 0,1V. Např. hodnota 120 odpovídá 12V. Pokud vstupní napětí překročí nastavený rozsah, vyskytne se chyba vstupního napětí.

SetMaximumTorque(int _maxTorque)

Nastaví maximální moment pohonu. Vstupní hodnota je typu int32. Rozsah hodnot je 0÷1023. Jeden dílek odpovídá 0,1%. Např. hodnota 1023 znamená, že pohon bude využívat 100% dostupného momentu. Při hodnotě 512 bude pohon využívat pouze 50% dostupného momentu.

SetStatustReturnLevel(int _SRL)

Nastaví hodnotu, jak se bude chovat vracet statusový paket. Vstupní hodnota je typu int32. Možné vstupní hodnoty jsou uvedené v Tab. 5.

SetAlarmLED(int _alarmLED)

Nastaví chyby, na které bude pohon reagovat blikáním LED diody. Seznam chyb je v *Tab. 6*.

SetAlarmShutdown(int _alarmShutdown)

Nastaví chyby, při kterých pohon vypne moment. Seznam chyb je v *Tab. 6*.

SetTorqueEnable(bool state)

Při hodnotě vstupu TRUE pohon drží moment. Při hodnotě vstupu FALSE pohon vypne moment.

SetLED(bool stav)

Ovládá kontrolní led na pohonu dynamixel. Vstup TRUE pro rozsvícení led, FALSE pro zhasnutí led.

SetGoalPosition(int CílováHodnota)

Nastaví cílovou hodnotu polohy pohonu. Vstupní hodnoty jsou typu int32. Rozsah hodnot je 0÷4095. Nejmenší dílek odpovídá 0,088°. Rozsah pohonu podle **Chyba! nalezen zdroj odkazů..**

SetMovingSpeed(int Rychlost)

Nastaví rychlost, se kterou se bude pohon pohybovat do cílové polohy. Vstupní hodnoty jsou typu int32. Vstupní hodnoty jsou typu int32. Rozsah hodnot je 0÷1023.

SetTorqueLimit(int Moment)

Nastaví velikost momentu, se kterým se bude pohybovat do cílové polohy. Vstupní hodnota je typu int32. Rozsah hodnot je 0÷1023. Jeden díl odpovídá 0,1% maximálního momentu.

SetGoalAcceleration(int zrychlení)

Nastaví cílovou hodnotu zrychlení pohonu. Vstupní hodnota je typu int32. Rozsah vstupních hodnot je 0÷254.

Convert_Derees_2_Position(float Stupně)

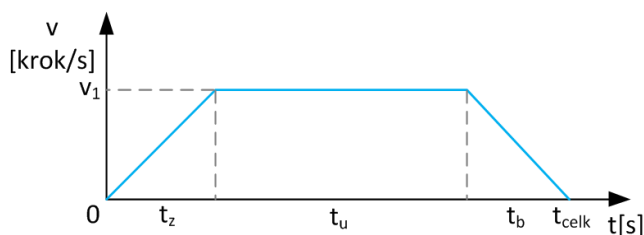
Převede úhlové stupně na jednotky polohy dynamixelu. Při přepočtu zahrnuje převod jednotlivých pohonů.

Convert_Position_2_Degrees(int Pozice)

Převede jednotky polohy dynamixelu na úhlové stupně. Při převodu zahrnuje převod jednotlivých pohonů.

CalcCasPohybu()

Pro současný dojezd všech článků je nutné vypočítat čas potřebný pro přesun z počáteční polohy do koncové polohy a podle nejpomalejšího času synchronizovat rychlosti ostatních pohonů.



Obr. 29 - Průběh rychlosti v závislosti na čase při pohybu kloubu, t_z - čas po který kloub zrychluje, t_u - čas po který jede objekt ustálenou rychlostí, t_b - čas při kterém objekt brzdí

Při výpočtu známe celkovou dráhu s_{celk} , rychlost v_1 a zrychlení a . Celkový čas t_{celk} přejezdu se vypočítá součtem časů potřebných pro zrychlený pohyb, ustálený a bržděný pohyb:

$$t_{celk} = t_z + t_u + t_b$$

kde t_z je čas zrychlení a t_b je čas brždění, t_u je čas ustáleného pohybu:

$$t_z = t_b = \frac{v_1}{a}$$

$$t_u = \frac{s_u}{v_1}$$

kde s_u je dráha ustáleného pohybu:

$$s_u = s_{celk} - s_z - s_b$$

kde s_z je dráha zrychlení a s_b dráha brždění:

$$s_z = s_b = \frac{1}{2} \cdot a \cdot t_z^2$$

Celkový čas pohybu potřebný pro přesun se vypočítá jako:

$$t_{celk} = 2 \cdot \frac{v_1}{a} + \frac{s_{celk} - 2 \cdot \left(\frac{1}{2} \cdot a \cdot \left(\frac{v_1}{a} \right)^2 \right)}{v_1}$$

$$t_{celk} = \frac{v_1}{a} + \frac{s_{celk}}{v_1}$$

Calc_Rychlost_Pohybu(double _ČasPohybu)

Po vypočtení časů potřebných pro pohyb všech článků je volající funkce porovná a vybere největší čas. Tento čas je použitý pro výpočet rychlosti článků tak aby všechny články dorazili do koncové polohy ve stejnou dobu. Rychlost se vypočítá:

$$v_1^2 - a \cdot t_{celk} \cdot v_1 + a \cdot s_{celk}$$

$$v_{1,2} = \frac{a \cdot t_{celk} \pm \sqrt{(-a \cdot t_{celk})^2 - 4 \cdot a \cdot s_{celk}}}{2}$$

Po dosazení vyjdou dva kořeny, použije se ten kořen, který je menší než původní rychlost.

3.2.3 Společné ovládací parametry pohonů Dynamixel MX

Oba pohony patří do stejné výrobní řady MX. Díky tomu mají stejné ovládací parametry. Pohon MX-28T nemá pět parametrů souvisejících s nastavením spotřeby proudu a s momentovým řízením.

Ovládací parametry pohonů jsou rozdělené do dvou typů pamětí. První typ paměti je EEPROM a druhý RAM. Po zapnutí a vypnutí pohonu data v EEPROM zůstávají uložené. Data v paměti RAM jsou vymazány.

V Tab. 7 jsou uvedené parametry chytrého pohonu dynamixel ukládané v paměti EEPROM.

Tab. 7 - Ovládací parametry registrů v paměti EEPROM [7][8]

Adresa	Originální název	Popis	Přístup	Výchozí hodnota
0 (0X00)	Model Number(L)	Low byte čísla modelu	R	29 (0X1D)
1 (0X01)	Model Number(H)	High byte čísla modelu	R	0 (0X00)
2 (0X02)	Version of Firmware	Informace o verzi firmware	R	-
3 (0X03)	ID	ID pohonu Dynamixel	RW	1 (0X01)
4 (0X04)	Baud Rate	Modulační rychlost pohonu Dynamixel	RW	34 (0X22)
5 (0X05)	Return Delay Time	Čas prodloužení návratu	RW	250 (0XFA)
6 (0X06)	CW Angle Limit(L)	Low byte úhlového limitu ve směru hodinových ručiček	RW	0 (0X00)
7 (0X07)	CW Angle Limit(H)	High byte úhlového limitu ve směru hodinových ručiček	RW	0 (0X00)
8 (0X08)	CCW Angle Limit(L)	Low byte úhlového limitu v protisměru hodinových ručiček	RW	255 (0XFF)
9 (0X09)	CCW Angle Limit(H)	High byte úhlového limitu v protisměru hodinových ručiček	RW	15 (0X0F)
11 (0X0B)	the Highest Limit Temperature	Vnitřní teplotní limit	RW	80 (0X50)
12 (0X0C)	the Lowest Limit Voltage	Spodní limit napětí	RW	60 (0X3C)
13 (0X0D)	the Highest Limit Voltage	Horní limit napětí	RW	160 (0XA0)
14 (0X0E)	Max Torque(L)	Low byte maximálního momentu	RW	255 (0XFF)
15 (0X0F)	Max Torque(H)	High byte maximálního momentu	RW	3 (0X03)
16 (0X10)	Status Return Level	Úroveň návratového statusu	RW	2 (0X02)
17 (0X11)	Alarm LED	LED při chybě	RW	36 (0X24)
18 (0X12)	Alarm Shutdown	Vypnutí při chybě	RW	36 (0X24)
20 (0X14)	Multi Turn Offset(L)	Low byte více otáčkového offsetu	RW	0 (0X00)
21 (0X15)	Multi Turn Offset(H)	High byte více otáčkového offsetu	RW	0 (0X00)
22 (0X16)	Resolution Divider	Dělič rozlišení	RW	1 (0X01)

V Tab. 8 jsou uvedené parametry chytrého pohonu dynamixel ukládané v paměti RAM.

Tab. 8 - Ovládací parametry registrů v paměti RAM [7][8]

Adresa	Originální název	Popis	Přístup	Výchozí hodnota
24 (0X18)	Torque Enable	Moment On/Off	RW	0 (0X00)
25 (0X19)	LED	LED On/Off	RW	0 (0X00)
26 (0X1A)	D Gain	Derivační zesílení	RW	0 (0X00)
27 (0X1B)	I Gain	Integrační zesílení	RW	0 (0X00)
28 (0X1C)	P Gain	Proporcionální zesílení	RW	32 (0X20)
30 (0X1E)	Goal Position(L)	Low byte cílové polohy	RW	-
31 (0X1F)	Goal Position(H)	High byte cílové polohy	RW	-
32 (0X20)	Moving Speed(L)	Low byte cílové rychlosti	RW	-
33 (0X21)	Moving Speed(H)	High byte cílové rychlosti	RW	-
34 (0X22)	Torque Limit(L)	Low byte cílového momentu	RW	ADD14
35 (0X23)	Torque Limit(H)	High byte cílového momentu	RW	ADD15
36 (0X24)	Present Position(L)	Low byte současné pozice	R	-
37 (0X25)	Present Position(H)	High byte současné pozice	R	-
38 (0X26)	Present Speed(L)	Low byte současné rychlosti	R	-
39 (0X27)	Present Speed(H)	High byte současné rychlosti	R	-
40 (0X28)	Present Load(L)	Low byte současného zatížení	R	-
41 (0X29)	Present Load(H)	High byte současného zatížení	R	-
42 (0X2A)	Present Voltage	Současné napětí	R	-
43 (0X2B)	Present Temperature	Současná teplota	R	-
44 (0X2C)	Registered	Registrovaná instrukce	R	0 (0X00)
46 (0X2E)	Moving	Pohon se pohybuje	R	0 (0X00)
47 (0X2F)	Lock	Zamknutí EEPROM	RW	0 (0X00)
48 (0X30)	Punch(L)	Low byte minimálního proudu	RW	0 (0X00)
49 (0X31)	Punch(H)	High byte minimálního proudu	RW	0 (0X00)
68 (0X44)	Current(L)	Low byte řídicího proudu	RW	0 (0X00)
69 (0X45)	Current(H)	High byte řídicího proudu	RW	0 (0X00)
70 (0X46)	Torque Control Mode Enable	Zapnutí momentového ovládání pohonu	RW	0 (0X00)
71 (0X47)	Goal Torque(L)	Low byte cílového momentu	RW	0 (0X00)
72 (0X48)	Goal Torque(H)	High byte cílového momentu	RW	0 (0X00)
73 (0X49)	Goal Acceleration	Cílové zrychlení	RW	0 (0X00)

3.2.4 Popis třídy Dynamixel.cs

Třída dynamixel.cs slouží pro zajištění komunikace mezi počítačem a pohonem Dynamixel.

dxl_initialize(číslo portu, baudrate)

Vytvoření komunikace mezi pohonem a řídicím počítačem. Vstup číslo komunikačního portu. Baudrate je nastavena podle Tab. 4.

dxl_terminate()

Ukončení komunikace mezi počítačem a pohony.

dxl_read_byte(ID, adresa)

Přečte jeden byt z dynamixelu. Vstupy ID pohonu a adresa parametru. Výstup hodnota daného registru 0÷254.

dxl_write_byte(ID, adresa, hodnota)

Zapíše jeden byt do dynamixelu. Vstupy ID pohonu, adresa a hodnota parametru.

dxl_read_word(ID, adresa)

Přečte dva byty z dynamixelu. Vstup ID pohonu a adresa prvního parametru. Výstup hodnota daného registru 0÷65 535.

dxl_write_word(ID, adresa, hodnota)

Zapíše dva byty do dynamixelu. Vstup ID pohonu, adresa prvního parametru a hodnota parametru.

3.2.5 Popis knihovny Dynamixel.dll

Dynamixel.dll je knihovna dodávána výrobcem chytrých pohonů. Knihovna obstarává komunikaci mezi počítačem a pohony.

Struktura složek knihovny Dynamixel.dll :

- /bin - ve složce se nacházejí dll soubory
- /import - ve složce se nacházejí nezbytné soubory pro programování
- /src - ve složce jsou zdrojové kódy dll knihovny
- /example - ve složce jsou příklady pro koncové uživatele

Knihovna je k dispozici v přílohách na CD přiloženém k diplomové práci nebo ke stažení na [6].

3.2.6 Popis třídy Dynamixel2Netduino

Třída Dynamixel2Netduino.cs má stejné veřejné funkce jako knihovna Dynamixel.cs obsahuje navíc pouze sériovou komunikaci mezi Netduinem2Plus a pohony dynamixel.

4 Závěr

V diplomové práci byla popsána mechanická konstrukce robotického ramene. Bylo sestaveno kinematické schéma ramene s popsanými jednotlivými klouby. Pohony ramene jsou tvořeny z chytrých pohonů dynamixel od firmy Robotis.

V práci byla popsána řídicí aplikace dodávaná výrobcem. Bylo ukázané ovládání aplikace, rozšíření aplikace o zásuvné moduly. Dále bylo popsáno jak uložit jednotlivé polohy ramene pro následné automatické projetí trasy.

Byla navržnuta vlastní ovládací třída pro článek a rameno. Na základě těchto tříd byla pro demonstraci funkčnosti navržnuta aplikace pro ovládání ve Windows a s pomocí Netduina. V práci jsou popsány vstupy a výstupy jednotlivých funkcí.

5 Zdroje a použitá literatura

- [1] Cyton Gamma 1500 Arm Specifications. In: *Lightweight Robotic Systems* | Robai [online]. Cambridge: Robai, ©2015 [cit. 2016-03-17]. Dostupné z: http://www.robai.com/assets/Cyton-Gamma-1500-Arm-Specifications_2015.pdf
- [2] Robai Cyton Gamma 1500. In: *Lightweight Robotic Systems* | Robai [online]. Cambridge: Robai, 2016 [cit. 2016-03-30]. Dostupné z: <http://www.robai.com/robots/robot/cyton-gamma-1500/>
- [3] SMPS2Dynamixel Adapter. In: *Trossen Robotics* [online]. Downers Grove, Illinois: Trossen Robotics, ©2016 [cit. 2016-03-30]. Dostupné z: <http://www.trossenrobotics.com/store/p/5886-SMPS2Dynamixel-Adapter.aspx>
- [4] USB2Dynamixel. In: *ROBOTIS* [online]. Seoul, South Korea: ROBOTIS, ©2010 [cit. 2016-03-30]. Dostupné z: http://support.robotis.com/en/product/auxdevice/interface/usb2dxl_manual.htm
- [5] Netduino Plus 2: technical specifications. In: *Netduino* [online]. New York: Secret Labs LLC, ©2010-2014 [cit. 2016-03-30]. Dostupné z: <http://www.netduino.com/netduinoplus2/specs.htm>
- [6] Dynamixel SDK for Windows. *ROBOTIS e-Manual* [online]. Seoul, South Korea: Robotis, 2010 [cit. 2016-05-16]. Dostupné z: http://support.robotis.com/en/software/dynamixel_sdk/usb2dynamixel/usb2dxl_windows.htm
- [7] MX-28T / MX-28R. *ROBOTIS e-Manual* [online]. Seoul, South Korea: Robotis, 2010 [cit. 2016-05-16]. Dostupné z: http://support.robotis.com/en/product/dynamixel/mx_series/mx-28.htm#Actuator_Address_2C
- [8] MX-64T / MX-64R. *ROBOTIS e-Manual* [online]. Seoul, South Korea: Robotis, 2010 [cit. 2016-05-16]. Dostupné z: http://support.robotis.com/en/product/dynamixel/mx_series/mx-64.htm#Actuator_Address_22

Seznam příloh

Obsah přiloženého CD:

Příloha A – robai_cd_soft.rar

Příloha B – dynamixelSDK.rar

Příloha C – RobaiWindows.rar

Příloha D – RobaiNetduino.rar

Diplomová práce ve formátech MS Word, PDF